AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A method in a computer system for determining resolution of attributes of a program, the method comprising:

analyzing said program prior to runtime, said analyzing prior to runtime including:

providing said ~~program, said~~ program having interactions, each interaction having commands with attributes;

identifying prior to runtime a sequence of interactions of the program; and for each interaction in the identified sequence,

for each command of the interaction,

for each input attribute of the command,

identifying prior to runtime an output attribute corresponding to the input attribute; and

indicating prior to runtime that the input attribute is resolved if the identified output attribute has been indicated as resolved, and indicating prior to runtime that the input attribute is not resolved if the identified output attribute has not been indicated as resolved; and

for each output attribute of the command, indicating prior to runtime that the output attribute is resolved.

2. (original) The method of claim 1 including reporting input attributes whose resolution is set to unresolved.

3. (original) The method of claim 2 including suppressing the reporting of input attributes that may be resolved by user input.

4. (original) The method of claim 2 including suppressing the reporting of input attributes of primitive types.

5. (currently amended) A method for verifying resolution of input parameters of functions of a computer program before executing the computer program, the method comprising:

verifying resolution of input parameters of functions of a computer program before executing the computer program, the verifying including:

providing, before executing the computer program, a path of execution of the computer program, the path of execution identifying a sequence of functions of the computer program; and

for each function identified in the provided path of execution, processing the function by

for each input parameter of the function, indicating before executing the program that the input parameter is resolved if a corresponding output parameter has been indicated as resolved as a result of a function in the path of execution having previously been processed; and

for each output parameter of the function, indicating before executing the program that the output parameter is resolved.

6. (original) The method of claim 5 wherein the computer program is a command-based application wherein the functions are methods of objects corresponding to the commands.

7. (original) The method of claim 6 wherein the commands are organized into interactions.

8. (original) The method of claim 6 wherein the parameters are attributes of the objects.

9. (original) The method of claim 8 wherein the objects have set and get methods for setting and getting attribute values.

10. (original) The method of claim 8 wherein the attribute values are set with an assignment statement.

11. (original) The method of claim 6 wherein each object has a perform method for performing a behavior associated with the command.

12. (original) The method of claim 5 wherein input and output parameters correspond when they have the same name.

13. (original) The method of claim 5 including creating a list of each parameter of each function processed, the list indicating resolution of the parameter, and including outputting an indication of resolution of each parameter based on the created list.

14. (original) The method of claim 5 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands, and wherein each command has a corresponding object with attributes.

15. (original) The method of claim 14 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

16. (original) The method of claim 14 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

17. (currently amended) A computer system, comprising:

means for verifying resolution of input parameters of functions of a computer program before executing the computer program, said means for verifying including:

means for selecting each function in execution order before executing the computer program; and means for processing each selected function before executing the computer program by for each input parameter of the function, indicating that the input parameter is resolved if a corresponding output parameter has been indicated as resolved as a result of a function of the computer program having previously been processed and for each output parameter of the function, indicating that the output parameter is resolved.

18. (original) The computer system of claim 17 wherein the computer program is a command-based application wherein the functions are methods of objects corresponding to the commands.

19. (original) The computer system of claim 18 wherein the commands are organized into interactions.

20. (original) The computer system of claim 18 wherein the parameters are attributes of the objects.

21. (original) The computer system of claim 20 wherein the objects have set and get methods for setting and getting attribute values.

22. (original) The computer system of claim 18 wherein each object has a perform method for performing a behavior associated with the command.

23. (original) The computer system of claim 17 wherein input and output parameters correspond when they have the same name.

24. (original) The computer system of claim 17 including means for creating a list of each parameter of each function processed, the list indicating resolution of the parameter, and including means for outputting an indication of resolution of each parameter based on the created list.

25. (original) The computer system of claim 17 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

26. (original) The computer system of claim 25 wherein each command is defined by a descriptor that optionally provides abasing for names of attributes.

27. (original) The computer system of claim 25 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

28. (currently amended) A computer system for processing each function of a computer program prior to runtime by for each input parameter of the function, determining prior to runtime whether a source of the input parameter would be resolved during execution of the computer program and for each output parameter of the function, indicating prior to runtime that the output parameter is resolved wherein output parameters are sources of input parameters.

29. (original) The computer system of claim 28 wherein the computer program is a command-based application wherein the functions are methods associated with objects corresponding to the commands.

30. (original) The computer system of claim 29 wherein the commands are organized into interactions.

31. (previously presented) The computer system of claim 29 wherein the parameters are attributes of the objects.

32. (original) The computer system of claim 31 wherein the objects have set and get methods for setting and getting attribute values.

33. (original) The computer system of claim 29 wherein each object has a perform method for performing a behavior associated with the command.

34. (original) The computer system of claim 28 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

35. (original) The computer system of claim 34 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

36. (original) The computer system of claim 34 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

37. (currently amended) A computer-readable medium containing instructions for controlling a computer system by a method comprising:

determining prior to runtime resolution of parameters of functions of a computer program, said determining including:

identifying a path of execution of the computer program prior to runtime, the path of execution having functions of the computer program;

for functions in the identified path of execution,

indicating prior to runtime that an input parameter of the function is resolved if a corresponding output parameter has previously been indicated as resolved; and

indicating prior to runtime that an output parameter is resolved.

38. (original) The computer-readable medium of claim 37 including indicating that an input parameter of a primitive type may be resolved by a user at runtime.

39. (original) The computer-readable medium of claim 37 wherein the computer program is a command-based application where the functions are methods associated with objects corresponding to the commands.

40. (original) The computer-readable medium of claim 39 wherein the commands are organized into interactions.

41. (original) The computer-readable medium of claim 39 wherein the parameters are attributes of the objects.

42. (original) The computer-readable medium of claim 41 wherein the objects have set and get methods for setting and getting attribute values.

43. (original) The computer-readable medium of claim 37 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

44. (original) The computer-readable medium of claim 43 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

45. (original) The computer-readable medium of claim 43 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.